## DRAWINGS

Please replace the attached Replacement Figure 3 Drawing, which was erroneously labeled "Fig. 1". This replacement drawing should replace "3/9" of the originally submitted drawings.

## REMARKS/ARGUMENTS

### *Claim Objections*

The Examiner has objected to claims 1, 6, and 7 as containing a misspelling. The Applicant has submitted an amendment to the claims to correct the misspellings. Accordingly, the Applicant respectfully requests that the claim objections be withdrawn.

### *Claim Rejections — 35 U.S.C. § 102 and 35 U.S.C. § 103*

The Examiner has asserted all of the identical claim rejections from the Office Action Dated May 14, 2008, as a result of the Examiner finding the Applicant's arguments in response to that Office Action unpersuasive.

The Applicant respectfully believes that the Examiner may be under some fundamental misunderstandings relating to the subject matter. Accordingly, the Applicant respectfully offers the following supplemental arguments and claim amendments and respectfully requests that the Examiner reconsider the Applicant's previous arguments in the Office Action dated May 14, 2008, which are hereby fully incorporated by reference into this response.

**Supplemental Arguments — Further Discussion of the Examiner-Cited References**

The main reference that the Examiner relies on, U.S. Patent No. 6,279,113 B1 to Vaidya (Vaidya), is a "classic" signature-based virus checker, of the type discussed in paragraph [0004] of the present application. In signature-based virus checker, files that are incoming to a computer system or network are analyzed to determine whether they include a particular "signature"; that is, a string of bytes corresponding to one of a library of signatures or strings of bytes stored by the virus checker. The signatures or strings of bytes stored in the library of the virus checker are those that are known to be used by known viruses that have previously been studied and analyzed (typically "by hand" by human operators).

Vaidya makes reference to the ideas of a "dynamic intrusion detection system" and the use of a "virtual processor" for detecting sequential, simple, and time-counter-based" attack signatures. However, this is not the disclosure of an intrusion detection system including means for receiving and storing one or more general rules, each of the general rules being representative of *the effect* on the computer system or network arising from a plurality of specific instances. On page 2 of the present Office Action, the Examiner refers to the disclosure in Vaidya of "attack

signature profiles" and states that these are descriptive of characteristics of known network security violations. Because of the Examiner's reliance on the "attack signature profiles" in Vaidya, it is important to understand exactly what the "attack signature profiles" are.

In any computing system, a message that is to be passed to the computer, or executed on it, will have a semantic intent. However, the message will be made up of syntax; that is, a string of bits and bytes. The attack signature profile of Vaidya is *not* a general rule that is representative of the *effect* or *semantic intent* on the computer of an intrusion or an attempted intrusion. Rather, it is a rule related to the syntax of a message that might be received by the computer. Referring to Figure 8 of Vaidya, there is shown an "attack signature profile 198". The description at column 9, line 46, through column 10, line 16, explains clearly what a typical attack signature profile might be composed of and what its function is. For example, at column 9, lines 47 through line 61, it is stated:

> "... wherein the expressions can be composed of search primitives 188, value primitives 190, and operators 192. In a preferred mode, the expressions also include keywords 193. An example of an expression might be as follows: (IP AND S1 and (V1>200)), wherein "IP" is a keyword referring to a packet utilising IP/TCP protocol, "S1" is a search primitive referring to user A, "AND" is a conjunctive operator, ">200" is an operator for indicating a value greater than 200, and "V1" is a value primitive referring to a packet length...".

In other words, what attack signature profile contains is a whole sequence of syntax or terms that is searched for within the received message. It is not searched for on the basis of the *effect* on the computer system, but rather based on the syntactic content of the message.

In contrast, in the present case, the intrusion detection system uses general rules that are representative of the *effect* on the computer system or network arising from a plurality of specific instances of intrusion or attempted intrusion.

The difference is significant. Referring to Figure 8 in Vaidya, the "expressions 194" within the attack signature profile have limited descriptive power. When a search strategy is based on looking for a sequence of syntax, then there is inevitably an infinite number of messages that will avoid the search but that could still cause damage. In addition, there will be messages that are caught but that are not harmful in intent. Thus, the presently claimed invention addresses this problem in a simple and robust manner by using general rules representative of the *effect* on the computer system of intrusion or attempted intrusion and *not* based on the syntactic content of any particular received message.

It is important to note that a signature-based virus checker looks at the bytes in the incoming file and determines whether there is a string of bytes that is the same as a signature or string of bytes stored in the library of the virus checker, and if so, typically triggers an alert that the incoming file is (potentially) dangerous. A signature-based virus checker is not concerned with the behavior of the incoming file or with the (potential) effect of the incoming file on the computer system or network. These signature-based virus checkers have a number of drawbacks as a consequence. First, they can lead to a high rate of "false positives": For example, the incoming file may actually be benign or innocuous and just happens to have a string of bytes that is the same as one of the signatures or strings of bytes stored by the virus checker in its library. Secondly, new viruses appear all the time, and it takes time for the human writers of the virus-checkers to analyze those new viruses and update the library of the virus-checker. This means that the computer system or network can be successfully attacked by a new virus in the time taken for the human operator to analyze the new virus and update the library. Also in existence are so-called polymorphic viruses which adapt and change their strings of bytes as they pass from one computer to another such that they do not contain a string of bytes corresponding to a signature stored in the library of the virus checker and therefore go undetected.

The presently claimed invention operates very differently from these signature-based virus checkers of the type disclosed by Vaidya. For example, please refer to paragraph [0019] of the present published application:

> The invention is based on the fact that it considers the functions that the computer input is likely to perform on the target computer rather than searching for a known form of attack inputs. The invention considers what the result of the input to the computer will be rather than what it looks like.

Vaidya is a classic example of a virus checker that "considers what the input to the computer looks like": As discussed above, Vaidya studies the bytes in the incoming file and compares those with signatures or strings of bytes stored by the virus checker in its library. In contrast, the presently claimed invention considers the *effect* of an incoming file; that is, the functions that the incoming file is likely to perform on the target computer or network. This is the central difference: The virus checker of Vaidya studies the incoming file itself, looking for strings of bytes that the virus checker knows or thinks are dangerous. Conversely, the presently claimed invention considers the (potential) *effect* of the incoming file on the computer system or network. This is an entirely different approach, and the Applicant respectfully submits that the presently claimed invention is neither anticipated nor obvious in view of Vaidya, either as a

stand-alone reference or in combination with any other Examiner-cited reference.

This difference is also delineated in paragraph [0012] of the present published application. The presently claimed invention uses *semantic* information about classes of attacks. In other words, whereas a virus checker like Vaidya uses a syntactic approach (that is, looking at the strings of bytes in an incoming file and comparing those with a library of signatures or strings of bytes), the presently claimed invention employs a *semantic* approach, which means that it looks at the meaning of sets of actions caused by an incoming file.

In short, the present invention uses an *effect*-based approach, using semantic information relating to the *effect* on the computer system or network arising from an incoming input file. Conversely, Vaidya, like many virus checkers, uses a syntactic approach, looking at the structure of the incoming input file itself. The presently claimed invention's new approach to the problem is much more flexible and robust. It is better able to cope with new and different viruses that do not have a structure (that is, a string of bytes) that has been seen previously, but nevertheless can bring about harmful activity to the computer system or network. That harmful activity is often of a type that has been seen previously and is characterized by the claimed "general" rules of the present invention.

In continuing to reject claims 1, 2, and 6 previously on file, the Examiner has explained that the Examiner regards the feature of those claims of "general rules being representative of characteristics associated with plurality of specific instances of intrusion or attempted intrusion" as being met by Vaidya's use-of-attack signature profiles. As explained, *supra*, and as supported by the original disclosure of the present published application (for example, paragraph [0019]), independent claims 1 and 6, and dependent claim 2, have been amended to refer to "each of said general rules being representative of *the effect* on the computer system or network arising from a plurality of specific instances of intrusion or attempted intrusion". The Applicant respectfully offers these amendments to clarify the distinction over Vaidya and similar signature-based virus checkers. In addition, independent claim 7 has been amended to similarly distinguish this claim from Vaidya such that the subject matter of claim 7 is not obvious in view of a combination of the Examiner-cited references Vaidya and "Applications of Inductive Logic Programming" (Bratko).

## ***Conclusion***

For all of the reasons stated herein, the Applicant has presented amendments, arguments, and facts that refute all of the Examiner's claim rejections. Therefore, the Applicant respectfully requests that all claim rejections be withdrawn and that a Notice of Allowance be issued.

Respectfully submitted,

/s/
Terrence M. Wyles
USPTO Reg. #61,035